Introduction to Big Data

Lesson 1.4 What if ... ?

Arthur Katossky & Rémi Pépin

Tuesday, March 29, 2023

What if ... ?

> Jane is a data analyst at Acme Corporation, a large online retailer.
>
> She needs to build a model to predict which customers are most likely to make a purchase in the next month so they can target them with personalized marketing campaigns. She thinks of a standard linear regression however the purchase data weighs 2To and is too large to fit in memory on her computer (16GB of RAM) or on the company's server (500 Go of RAM).

Example adapted form suggestion by GPT-3.5.

What if data is too big to fit in memory ?

1. Buy more / bigger memory cards
   Easy but expensive. Not the best solution
2. Do less (filter, select, sample)
   Do you really need all this data ? Can statistic help you ?
3. Do things in chunks / stream
   Need to rethink the process. Or to use exotic libraries.
4. Rent one virtual machine in the cloud
   Need to store the data at the same service provider.
5. Use a computing service
   For simple uses (exploratory statistics, mainstream ML), cloud providers have your back

What if ... ?

> Sarah works at Global Logistics Inc., a transportation company. There, she analyzes GPS data from the company's fleet of vehicles to optimize routes and fuel consumption.
>
> The data is stored on a dedicated server, where each new datum is appended to a gigantic text file. Some day, Sarah faces a single 5 Go file, but her workstation runs under Windows with FAT32 file system and she cannot open it anymore. When she borrows her manager's computer and tries to open the file from the file explorer, it takes 15 minutes just to open and the whole operating system is slowed down.

What if your file is too big for your local file system ?

1. Update file system
   No more FAT32
2. Import less data (filter, select, sample)
   For a small experimentation, you don't need 2To of data
3. Cut file into pieces and process in chunks
   For instance 1 file per day
   (many data formats are text under the hood)
4. If your data are structured, even so slightly, move to a database
   That's what databases are made for
5. Rent a remote cloud-based service, they will abstract the problem away from you
   Like AWS S3, Azure Bloc Blobs, GCP Storage

What if your file is too big for your local file system ?

Q: How much data can I store?

The total volume of data and number of objects you can store are unlimited. Individual Amazon S3 objects can range in size from 1 byte to 5 terabytes. The largest object that can be uploaded in a single PUT is 5 gigabytes. For objects larger than 100 megabytes, customers should consider using the Multipart Upload capability.

What if your file is too big for your local file system ?

General Availability: Larger Block Blobs in Azure Storage

Posted on December 22, 2016

Michael Hauss, Program Manager, Azure Storage

Azure Blob Storage is a massively scalable object storage solution capable of storing and serving tens to hundreds of petabytes of data per customer across a diverse set of data types including media, documents, log files, scientific data and much more. Many of our customers use Blobs to store very large data sets, and have requested support for larger files. The introduction of larger Block Blobs increases the maximum file size from 195 GB to 4.77 TB. The increased blob size better supports a diverse range of scenarios, from media companies storing and processing 4K and 8K videos to cancer researchers sequencing DNA.

What if your file is too big for your local file system ?

Google Cloud

- There is a maximum size limit of 5 TB for individual objects stored in Cloud Storage.

What if ... ?

> Kim is data scientist at West Coast Energy. The company is launching a new project where they plan to analyse satellite imagery to identify potential locations for new wind farms.
>
> The high-resolution imagery covers a large area and generates huge amounts of data that cannot be stored on an average desktop computer.

What if data is too big to fit on disk ?

1. Store less (filter, select, sample)

   Especially for exploratory projects
2. Buy bigger physical disk

   (Easier on desktops)
3. Buy your own storage server

   Distribution is transparent. Local replication possible.
4. Rent storage space (file space or database) in the cloud

   It is transparently distributed and replicated over a cluster

   but there are privacy or sovereignty concerns!
5. Distribute data on an organisation-hosted cluster

   Quite technical. Local replication possible.

What if ... ?

> Amin works at an Environmental research organization called Noah's Ark. The organisation is running simulations of climate models to predict future climate patterns and identify potential areas of concern for conservation efforts.
>
> The models are borrowed from a preliminary project from a partner university and have a manageable size (a few GB) but executing just one simulation at a low resolution, on a regional scale and in the short run already takes days on Amin's laptop (32 GB of RAM). However, his manager asked for multiple variants based on the IPCC latest scenarii.

What if computation takes ages ?

1. Do less: filter, select, sample
   - Especially in development stages
   - Processing is the most costly aspect of statistics in production
   - ... both in dollars and environmentally-speaking
2. Profile your code
   - How does it scale ? Where are the bottlenecks?
   - Then focus on these parts
3. Do less: less tests, less formatting
4. Use parallelism locally
   - Works on both CPUs and GPUs
5. Go low-level: compile, use C or C++, use command shells ... or build chips!

What if computation takes ages ?

6. Buy processors with faster or more cores

7. Move computation closer to the data source in a first stage and use the ouputs for final computation

This redirects the computation load towards the data source

8. Rent low-level infrastructure (virtual machines) or high-level computing service (such AWS SageMaker) over the cloud

Same privacy or sovereignty concerns as before

9. Avoid i/o or network operations

10. Use a supercomputer, a grid or a cluster

Developing your own cluster solution may be quite challenging

... using one less so

What if ... ?

What if computation / storage is too expensive ?

1. Store or compute less (filter, select, sample)
2. Consider cloud computing

   Not always cheaper, though!
3. Be careful with databases requests

   (When you pay on read)
4. Go from RAM (expensive) to disk (cheap)
5. Use smaller but more computing units

   Scientific grids, edge computing, fog computing, etc.

   Technically challenging

What if ... ?

What if data i/o is too fast ?

1. For computing: pipelining (overlapping chunk processing), stream (never-ending one-by-one processing)
2. For storage: fast databases

What if ... ?

What if data i/o is too varied ?

1. Dedicated file systems and databases

This is ultimately not a computation issue.